

## CONTEXTE (StatCode)



**StatCode** est une application qui permet la gestion des sessions de code de la route afin de garder les notes pour pouvoir ensuite faire des statistiques sur l'avancée de l'utilisateur:

Public visé : Personnes voulant passer l'examen du code de la route, Une session se présente sous la forme d'une note sur 40, qu'une date inférieure à celle d'aujourd'hui, d'un niveau de difficulté ressenti par l'utilisateur (noté sur 5 pts) sur la globalité des questions faite pendant la session, une note/description si l'utilisateur veut se rappeler d'une information importante ou savoir comment s'est passé la session et finalement une couleur dédiée en fonction de la note ( rouge:0-24 orange:25-34 vert:35-40)

Il y a bien sûr la possibilité d'ajouter une nouvelle session dans le menu principal pour un ajout rapide et une page consacrée à la gestion des sessions qui permettra dès le supprimer ou les modifier si une erreur s'est présenté.

Une page historique est consacrée à la partie master-detail en affichant toutes les informations des sessions dans sa liste de sessions

Une page Statistiques permet d'avoir plus d'informations sur les sessions passées:

- Diagramme en bâton pour chaque session faite pour voir le rapport de nombre de fautes sur le total de 40 points
- Barre de progression (si elle est remplie, l'utilisateur sera fin prêt pour passer le code)
- Moyenne Générale de toutes les sessions enregistrées
- Moyenne sur les 5 dernières notes des sessions pour ne pas prendre en compte les sessions ratées auparavant
- Nombres de jours restants avant l'examen final (l'utilisateur peut ajouter cette date en cours de période de code)

## PERSONAS

Persona 1 : Guillaume Valto

“ Je suis perdu. Je ne sais pas où j'en suis dans ma progression au code la route. ”

Guillaume est au lycée, il peine à allier cours et code. Il n'a pas le temps de consacrer beaucoup de temps à sa progression sur son code de la route.

Age : 16 ans

Situation : Étudiant au lycée

Revenus : Aucun

Profil Technique : Ordinateur Fixe DELL avec Windows 10

Smartphone Android

Internet : 1h / jour ( études )

2h / jour ( personnel : Réseaux Sociaux / Youtube )

Il sait bien utiliser Internet

Sites préférés : Facebook / Youtube / ENT

Applications préférées : Youtube / Snapchat / Instagram / Youtube

Persona 2 : Helene

“ Je suis très occupée par mon travail, et je n'ai pas le temps de m'occuper de mon avancée sur le passage du code de la route.

J'habite à Paris, je vais déménager, j'ai besoin de passer mon code de la route assez rapidement pour ensuite avoir mon permis. ”

Hélène est secrétaire et a juste le temps de se rendre à son auto-école pour faire une session de code par jour.

Elle fait sa session de code le matin avant d'aller au travail ou le soir en revenant de son travail.

Âge : 36 ans

Situation professionnelle : Secrétaire administrative

Situation personnelle : Mariée / 2 enfants ( 5 ans / 9 ans )

Revenus : 26500€ / an

Profil Technique : Ordinateur Portable HP avec Windows 10

Smartphone Iphone 10 avec iOS

Internet : 5h / jour pour son travail

1h30 par jour pour son usage personnel : Informations,  
réseaux sociaux

Elle sait bien utiliser Internet.

Sites préférés : Marmite / Boite Mail / Facebook

Applis préférées : Candy Crush Saga / Le Monde

Persona 3 : Stéphane Loustic

“Je suis étranger, je fais une thèse et j’habite en France depuis peu. Il me faut donc refaire tous mes papiers donc mon permis de conduire. Pour cela il faut que je repasse mon permis de conduire.”

Stéphane est en train de faire une thèse et doit refaire son permis en France car il vient d’arriver dans le pays. Mais il n’a pas le temps de se consacrer pleinement au code de la route.

Âge : 42 ans

Situation Professionnelle : Enseignant - Chercheur

Situation Personnelle : Divorcé / 1 enfant ( 13 ans )

Profil Technique :      Ordinateur Portable Lenovo avec Windows 10

Smartphone Samsung avec Android

Internet :              8h / jour pour son travail

2h / jour pour son usage personnel ( Informations,

Réseaux Sociaux.

Il sait très bien utiliser Internet.

Sites préférés : Google Chrome / ENT / Microsoft Teams

Applications préférées : Twitter / Facebook

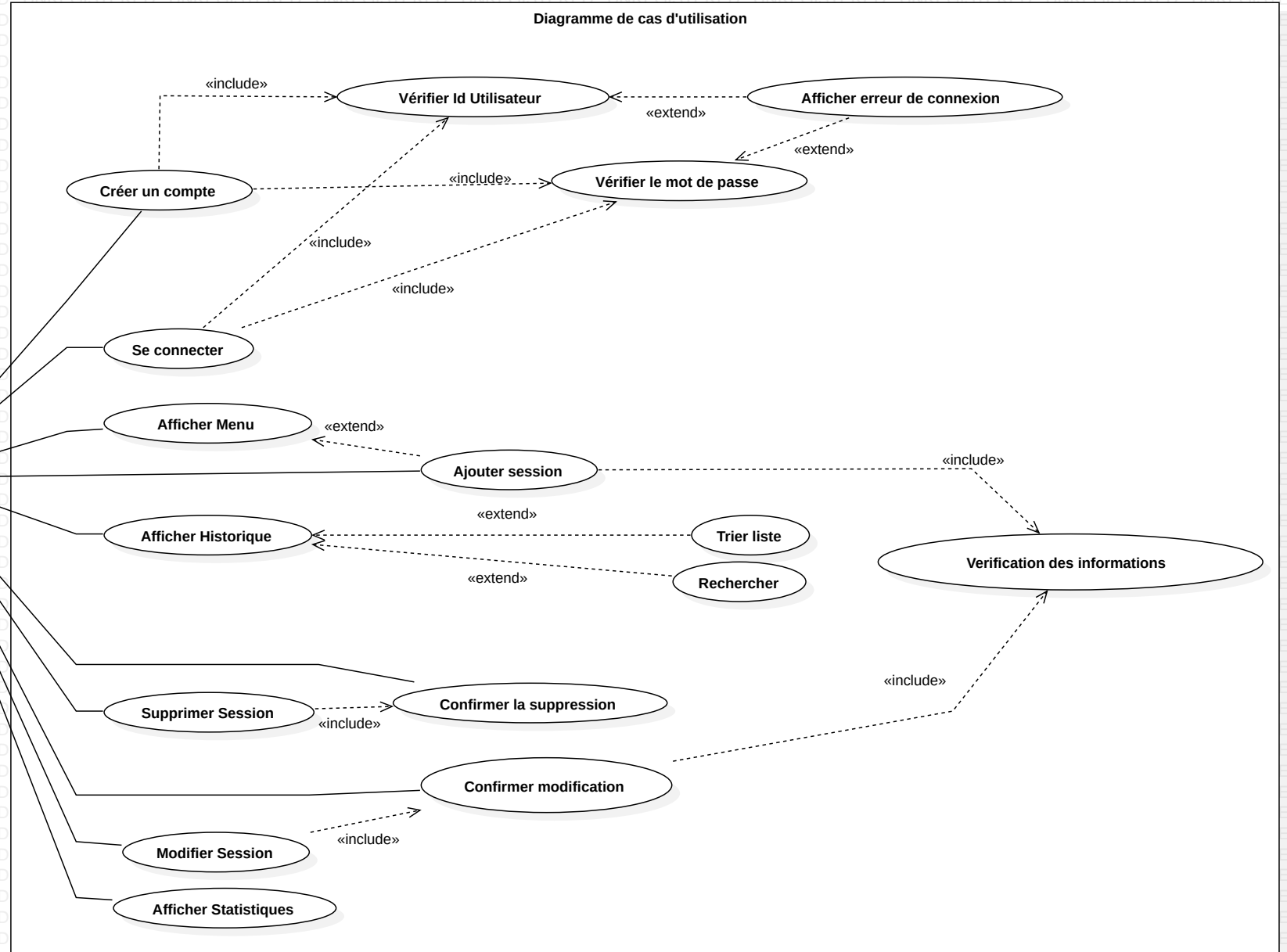
## USER-STORIES

Mercredi , début d'après-midi. Guillaume va à l’auto-école pour faire une session de code. Mais comme il n’y va pas régulièrement, il ne sait pas où il en est dans sa progression. Cela fait environ 4 mois qu’il a commencé et il ne sait pas quelle est sa moyenne, ni combien de sessions il a fait. Il en a aucune idée. Mais heureusement, le soir quand il va rentrer chez lui il pourra consulter l’application StatCode qui lui donnera toutes les statistiques liées à sa progression au code de la route.

Mardi matin 5h30. Hélène est prête à partir de chez elle. Elle part de chez elle à pied pour se rendre à l'auto-école. Elle arrive à l'auto-école à 5h53. L'auto-école ouvre à 6h00. 6h00, début de la session de code. Elle fait sa session de code et fait 7 fautes. Il est désormais 7h00 et elle doit aller à son lieu de travail. Elle arrive à son travail à 7h25 et elle commence le travail à 7h30. A 9h00, elle a une pause, elle en profite pour consulter l'application StatCode sur ordinateur pour voir les statistiques ainsi que sa progression. Elle y ajoute la session qu'elle a fait avant de venir au travail. Elle peut désormais voir ses statistiques qui sont mises à jour instantanément. Elle voit notamment que l'application lui indique qu'elle peut passer l'examen du code de la route au vu de ses derniers résultats.

Jeudi après-midi. Stéphane va au code dès que possible. Il n'a pas le temps de se concentrer sur sa progression du code de la route. Ce soir, après le travail il va aller à l'auto-école pour faire une session.

Jeudi soir. Stéphane vient de rentrer chez lui. Comme il a 10 minutes devant lui, il s'empresse d'aller sur son ordinateur pour voir ses statistiques. Il rentre la session qu'il vient de faire et peut ainsi voir sa nouvelle moyenne en temps réel. Il est désormais fixé sur sa progression du code de la route et peut se faire une idée de la date d'examen.



## Documentation du diagramme de cas d'utilisation (StatCode)

### Cas « Se Connecter »

Nom	Se Connecter
Objectif	Permettre à un utilisateur de se connecter à l'application
Acteur principal	Utilisateur
Conditions initiales	L'utilisateur doit avoir lancé l'application au préalable. L'utilisateur doit se connecter avec son compte utilisateur.
Scénario d'utilisation	L'utilisateur lance l'application. Il entre son identifiant Utilisateur. L'application vérifie l'identifiant utilisateur par le cas « Vérification de l'id Utilisateur ». (Si l'id utilisateur n'est pas correct, conditions de fin 2) L'utilisateur rentre son mot de passe. L'application vérifie le mot de passe par le cas « Vérifier le mot de passe ». (Si le mot de passe n'est pas correct, condition de fin 2) S'il n'y a pas d'erreur, condition de fin 1
Conditions de fin	<ol style="list-style-type: none"> <li>1) L'identification s'est passée sans problème, l'utilisateur est connecté sur son compte personnel.</li> <li>2) L'id utilisateur ou le mot de passe ne sont pas correct, ressaisir les informations.</li> </ol>

### Cas « Créer un compte »

Nom	Créer un compte
Objectif	Permettre à un nouvel utilisateur de se créer un nouveau compte
Acteur principal	Utilisateur
Conditions initiales	L'utilisateur doit avoir lancé l'application au préalable. L'utilisateur doit cliquer sur « Créer un compte »
Scénario d'utilisation	L'utilisateur lance l'application. Il clique sur « Créer un compte ». L'application vérifie le format de l'identifiant saisi par le cas « Vérification de l'id Utilisateur ». (Si l'id utilisateur n'est pas correct, conditions de fin 2) L'utilisateur rentre son nouveau mot de passe. L'application vérifie le format du mot de passe par le cas « Vérifier le mot de passe ». (Si le mot de passe n'est pas correct, condition de fin 2) S'il n'y a pas d'erreur, condition de fin 1
Conditions de fin	<ol style="list-style-type: none"> <li>1) Le format de l'identifiant et du mot de passe sont corrects, l'utilisateur est connecté sur son compte personnel.</li> <li>2) L'id utilisateur ou le mot de passe ne sont pas correct, ressaisir les informations.</li> </ol>

## Documentation du diagramme de cas d'utilisation (StatCode)

### Cas « Afficher Menu »

Nom	Afficher Menu
Objectif	Permettre à l'utilisateur de voir le menu.
Acteur principal	Utilisateur
Conditions initiales	L'utilisateur doit avoir lancé l'application au préalable. L'utilisateur doit être connecté sur son compte.
Scénario d'utilisation	L'utilisateur lance l'application. Il clique sur « Créer un compte » / Se Connecter. Une fois que l'utilisateur a cliqué sur « Se connecter » ou « Créer un compte », alors il se trouve sur le Menu, aussi appelé « Page d'accueil ».
Conditions de fin	

### Cas « Ajouter session »

Nom	Ajouter session
Objectif	Permettre à un utilisateur d'ajouter une session de code.
Acteur principal	Utilisateur
Conditions initiales	L'utilisateur doit avoir lancé l'application au préalable. L'utilisateur doit être connecté.
Scénario d'utilisation	L'utilisateur lance l'application. L'utilisateur se connecte ou crée un compte, si non condition de fin 2. L'application s'ouvre sur le compte de l'utilisateur, il voit le menu et clique sur le bouton « Ajouter une session ». Une nouvelle fenêtre s'ouvre et l'utilisateur saisi les informations de la session, la vérification de la saisie des informations se fait par le cas « Vérification des infos d'ajout » (Appel de la condition de fin 3).
Conditions de fin	<ol style="list-style-type: none"> <li>1) Le format des informations sont corrects, la session est ajoutée a son compte personnel.</li> <li>2) L'utilisateur n'arrive pas a se connecter, alors retour au cas « Se connecter » ou « Créer un compte ».</li> <li>3) Le format des informations est incorrect ou certaines informations n'ont pas été saisies, ressaisir les informations.</li> </ol>

## Documentation du diagramme de cas d'utilisation (StatCode)

### Cas « Afficher Historique »

Nom	Afficher Historique
Objectif	Permettre à un utilisateur de consulter son historique des sessions enregistrées.
Acteur principal	Utilisateur
Conditions initiales	L'utilisateur doit avoir lancé l'application au préalable. L'utilisateur doit être connecté.
Scénario d'utilisation	L'utilisateur lance l'application. L'utilisateur se connecte ou crée un compte, si non condition de fin 2. L'application s'ouvre sur le compte de l'utilisateur, il voit le menu et clique sur le bouton « Historique ». Un nouvel onglet s'ouvre et l'utilisateur peut ainsi voir les informations de toutes ses sessions enregistrées. Il est possible de trier la liste des sessions par note. Il est également possible de rechercher une certaine session souhaitée
Conditions de fin	1) L'onglet « Historique » s'affiche correctement. 2) L'utilisateur n'arrive pas à se connecter, alors retour au cas « Se connecter » ou « Créer un compte ».

### Cas « Supprimer Session »

Nom	Supprimer session
Objectif	Permettre à un utilisateur de supprimer une session.
Acteur principal	Utilisateur
Conditions initiales	L'utilisateur doit avoir lancé l'application au préalable. L'utilisateur doit être connecté.
Scénario d'utilisation	L'utilisateur lance l'application. L'utilisateur se connecte ou crée un compte, si non condition de fin 2. L'application s'ouvre sur le compte de l'utilisateur, il voit le menu et clique sur le bouton « Historique ». Un nouvel onglet s'ouvre et l'utilisateur peut ainsi voir les informations de toutes ses sessions enregistrées. Il peut ainsi supprimer une session de code. Une fenêtre de confirmation pour la suppression s'ouvre avec le cas « Confirmer suppression » et l'utilisateur indique s'il veut bien supprimer la session sélectionnée ou non, sinon condition de fin 3)
Conditions de fin	1) L'onglet « Historique » s'affiche correctement. 2) L'utilisateur n'arrive pas à se connecter, alors retour au cas « Se connecter » ou « Créer un compte ». 3) Si l'utilisateur ne veut pas supprimer, alors retour à l'onglet « Historique »



## Documentation du diagramme de cas d'utilisation (StatCode)

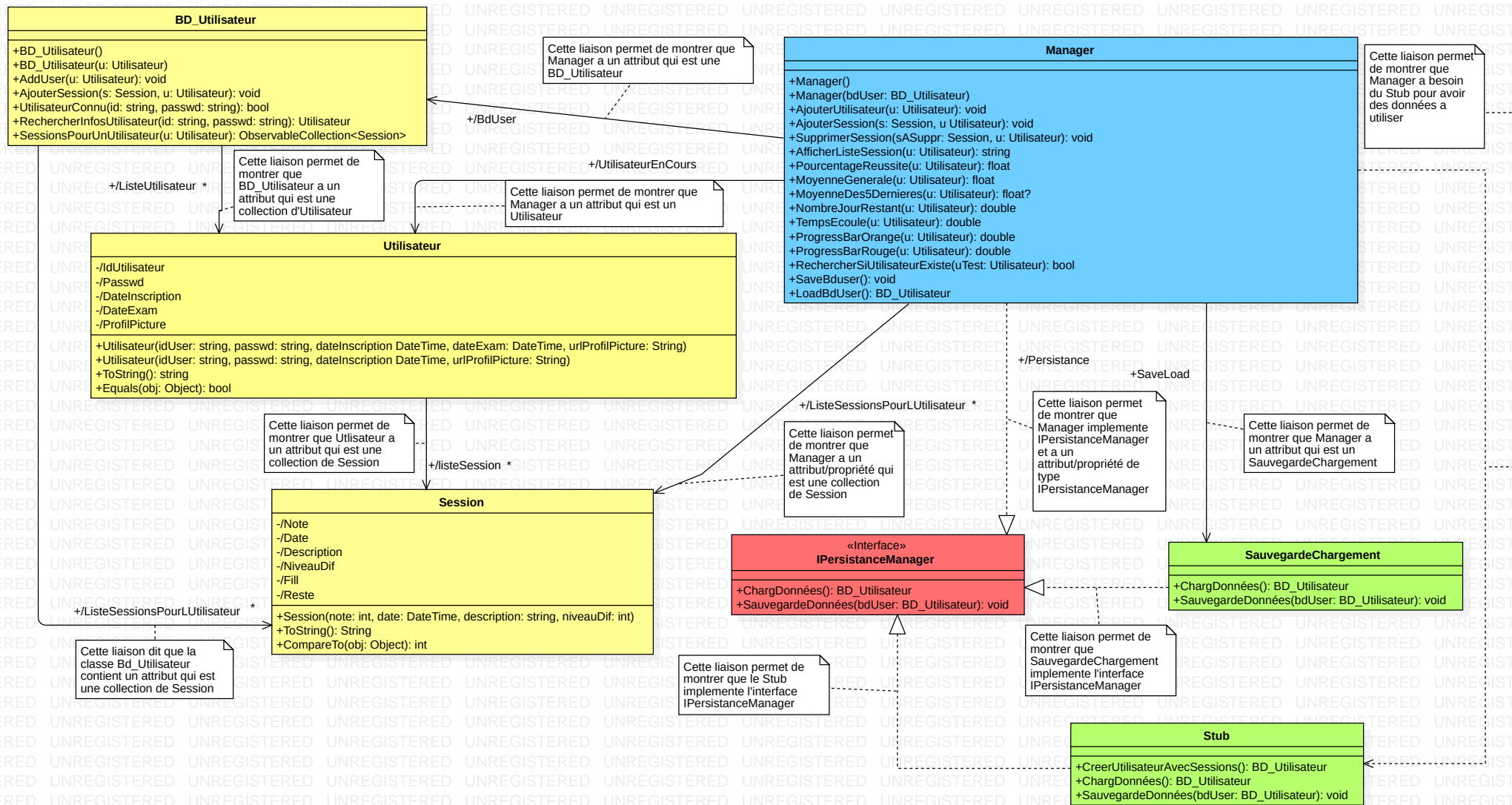
### Cas « Modifier Session »

Nom	Modifier session
Objectif	Permettre à un utilisateur de modifier une session.
Acteur principal	Utilisateur
Conditions initiales	L'utilisateur doit avoir lancé l'application au préalable. L'utilisateur doit être connecté.
Scénario d'utilisation	L'utilisateur lance l'application. L'utilisateur se connecte ou crée un compte, si non condition de fin 2. L'application s'ouvre sur le compte de l'utilisateur, il voit le menu et clique sur le bouton « Historique ». Un nouvel onglet s'ouvre et l'utilisateur peut ainsi voir les informations de toutes ses sessions enregistrées. Il peut ainsi modifier les données d'une session de code. Une fenêtre de confirmation pour la modification s'ouvre avec le cas « Confirmation suppression » et l'utilisateur indique s'il veut bien modifier la session sélectionnée ou non, sinon condition de fin 3) La vérification du format des données modifiées sont modifiées par le cas « Vérification modification », si non condition de fin 4)
Conditions de fin	<ol style="list-style-type: none"> <li>1) L'onglet « Historique » s'affiche correctement.</li> <li>2) L'utilisateur n'arrive pas à se connecter, alors retour au cas « Se connecter » ou « Créer un compte ».</li> <li>3) Si l'utilisateur ne veut pas modifier les données d'une session, alors retour à l'onglet Historique.</li> <li>4) Format des nouvelles données saisies mauvaises, retour au cas « Modifier session »</li> </ol>

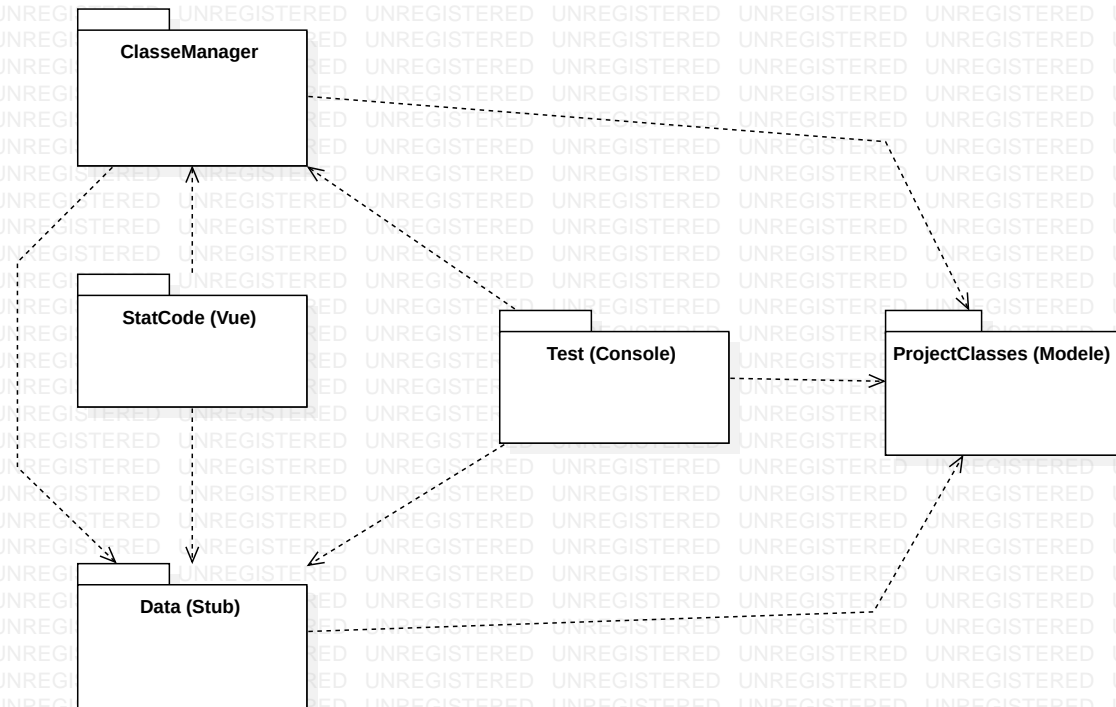
### Cas « Afficher Statistiques »

Nom	Afficher Statiques
Objectif	Permettre à un utilisateur de consulter ses statistiques.
Acteur principal	Utilisateur
Conditions initiales	L'utilisateur doit avoir lancé l'application au préalable. L'utilisateur doit être connecté.
Scénario d'utilisation	L'utilisateur lance l'application. L'utilisateur se connecte ou crée un compte, si non condition de fin 2. L'application s'ouvre sur le compte de l'utilisateur, il voit le menu et clique sur le bouton « Statistiques ». Un nouvel onglet s'ouvre et l'utilisateur peut ainsi consulter les statistiques de toutes ses sessions enregistrées.
Conditions de fin	<ol style="list-style-type: none"> <li>1) L'onglet « Statistiques » s'affiche correctement.</li> <li>2) L'utilisateur n'arrive pas à se connecter, alors retour au cas « Se connecter » ou « Créer un compte ».</li> </ol>

## Model

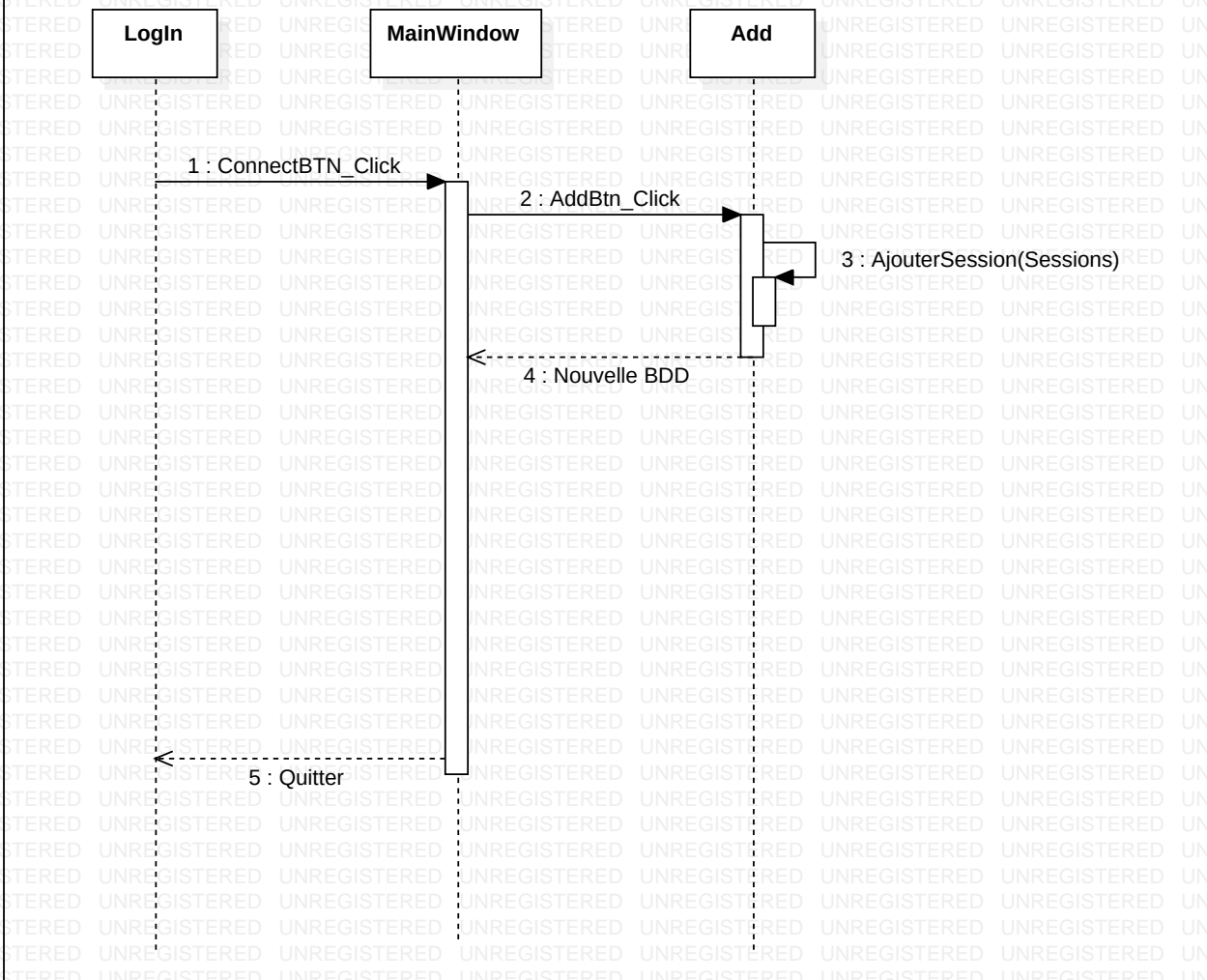


## Model : Package Diagram



Ce diagramme de Paquetage permet de savoir quels liens il y a entre les différents paquetages. Par exemple ClasseManager a besoin de Data et ProjectClasses pour fonctionner.

interaction SequenceDiagram1



Ce diagramme de séquence nous permet de savoir comment se déroule un ajout dans la base de donnée  
Dans un premier temps, l'utilisateur se connecte en appuyant sur le bouton ConnectBTN\_Click puis ajoute une session qu'il a fait en appuyant sur le bouton AddBtn\_Click  
La méthode AjouterSession(Session) permet de confirmer l'ajout de la session et renvoie la nouvelle base de donnée

Se connecter

Nom

Igor

Mot de passe

\*\*\*\*\*

Se connecter

Créer un compte

Permettre à un utilisateur de se connecter sur son compte personnel en renseignant son nom d'utilisateur et son mot de passe afin d'avoir ses résultats personnalisées.

Possibilité de créer un compte si l'utilisateur n'a pas de compte personnel, en cliquant sur le bouton "Créer un compte".

Nom

Igor

Mot de passe

\*\*\*\*\*

Confirmer mot de passe

\*\*\*\*\*

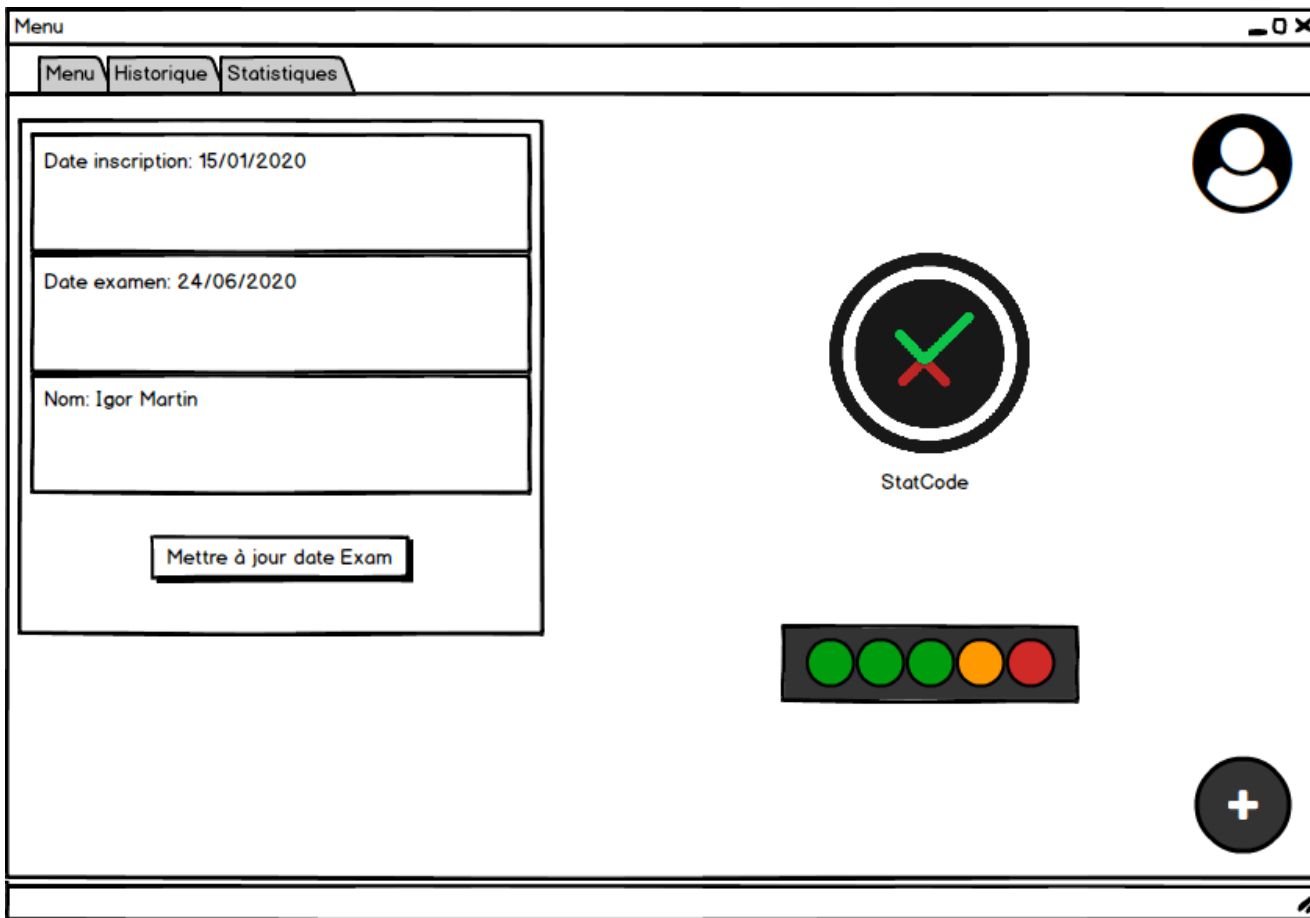
Photo de profil:

Choisir ...



Créer & Se connecter

Permettre a un nouvel utilisateur de créer un compte en saisissant un nom d'utilisateur et un mot de passe puis une confirmation du mot de passe.



Permet à l'utilisateur connecté d'avoir un aperçu global des sessions effectuées.

Dans la partie gauche de la page il y a son nom, sa date d'inscription à l'auto-école et sa date d'examen potentielle au code la route.

A droite, il y a un graphique qui représente la progression de l'utilisateur. Sur l'axe des ordonnées, il y a ses notes obtenues et sur l'axe des abscisses, il y a le temps.

Le bouton "+" en bas à droite permet d'ajouter une nouvelle session.

Ajouter

Note

21


Niveau difficulté

5

Description

Je ferais mieux la prochaine fois

01 / 04 / 2020



Ajouter

Permet l'ajout d'une nouvelle session de code en entrant la nouvelle note, le niveau de difficulté ressenti par l'utilisateur, la description, puis enfin la date qu'à eu lieu la session.



Historique

Menu

Historique

Statistiques

Igor

Note

Date

21/03

25/03

26/03

30/03

01/04

21

21 / 03 / 2020

Description:

Quelques erreurs d'inattention

Niveau de difficulté :

★★★★☆

Modifier

Supprimer

Suppression

Etes-vous sur de vouloir supprimer la session sélectionnée?

Non

Oui

Permettre à l'utilisateur de consulter son historique des sessions effectuées

Il est possible de rechercher une session en particulier.

Permet d'afficher la liste de toutes les sessions dans le master, puis dans le detail, tout les détails concernant la session sélectionnée

Le diagramme en camembert représente la note de la session. Il y a la date de la session , la description et le niveau de difficulté associés à la session selectionnée.

Note

26

Niveau difficulté



5

Description

Je ferais mieux la  
prochaine fois

01 / 04 / 2020



Modifier

Permet de modifier les données  
d'une session.

Saisir la nouvelle note , le  
nouveau niveau de difficulté, la  
nouvelle description et la  
nouvelle date.

Mise a jour date exam

Choisir Date

01 / 04 / 2020

Confirmer

Permet de modifier ou de renseigner la date de l'exam que l'auto-école le lui a communiqué



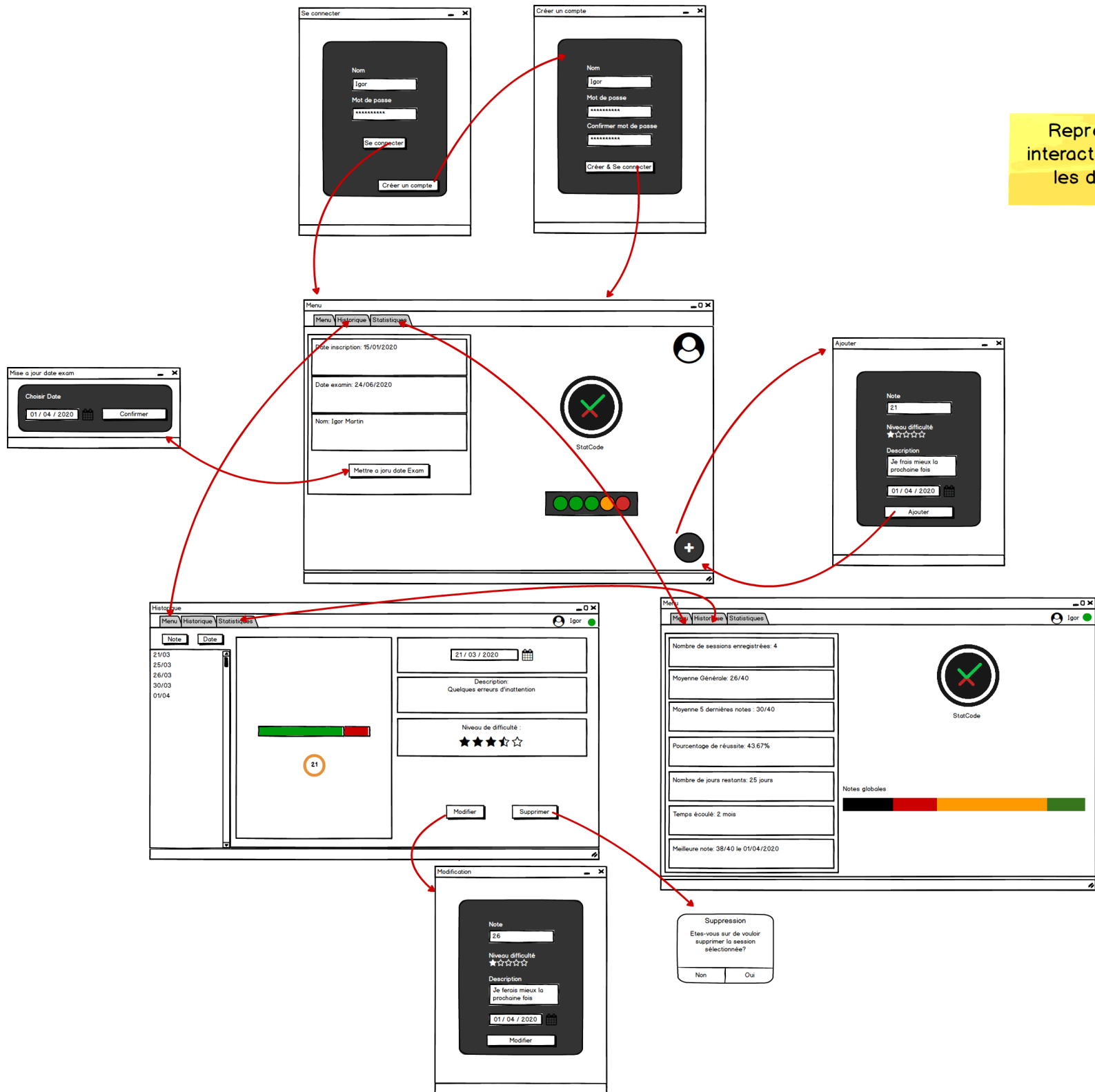
Permet d'afficher toutes les statistiques concernant les sessions

Dans le master, toutes les informations à propos de l'utilisateur.

Dans le détail :  
Progressions 5 dernières sessions:  
Si la barre de progression est arrivée à 100% c'est que l'utilisateur est prêt à passer l'examen final.

Notes globales:  
Représente la part de chaque couleur de la note.  
ex: si il y a plus de jaune, cela veut dire que l'utilisateur a eu plus de note entre 26 et 34 que le reste des résultats.

Représente toutes les interactions possibles entre les différentes pages



Partie Persistance :

Nous avons utilisé la persistance avec un fichier binaire dans notre projet. Cela nous permet d'enregistrer et stocker les données en permanence dans un fichier binaire (BinaryFormatter) pour pouvoir les récupérer au prochain lancement de l'application.

Pour pouvoir utiliser la persistance, nous avons créé une interface IPersistanceManager qui contient deux méthodes : 'BD\_Utilisateur ChargDonnées()' et 'void SauvegardeDonnées(BD\_Utilisateur bdUser)'

Les méthodes ne sont pas définies dans l'interface, mais dans une autre classe nommée SauvegardeChargement qui implémente la classe IPersistanceManager

Cette classe SauvegardeChargement permet de définir ce que font les deux méthodes

La méthode BD\_Utilisateur ChargDonnées() permet de charger les données qui sont dans le fichier binaire pour pouvoir récupérer les données qui sont à l'intérieur au lancement de l'application

La méthode void SauvegardeDonnées(BD\_Utilisateur bdUser) permet de sauvegarder les données dans le fichier binaire.

Pour pouvoir utiliser la persistance dans notre projet, on a implémenté l'interface IPersistanceManager dans la classe Manager et redéfini les deux méthodes de l'interface pour les faire correspondre avec les classes de notre projet

Pour créer les données qui seront utilisées dans l'application, il faut implémenter l'interface IPersistanceManager dans la classe Stub et redéfinir les deux méthodes de l'interface.

La méthode de chargement des données dans le Stub permet de créer les collections de données par défaut qui seront utilisées pour tester l'application.

La méthode de sauvegarde des données dans le Stub permet d'afficher un message de Debug quand une sauvegarde est demandée.

Pour pouvoir charger les données du stub dans le fichier, il faut instancier le manager avec BD\_Utilisateur créer dans le stub (dans le fichier App.xaml.cs)

Ensuite, une fois que le fichier est rempli, nous pouvons lancer le projet avec le contracteur par défaut de manager qui permet de lire les données du fichier

A chaque modification ou ajout, l'application fait appelle a la sauvegarde

Pour pouvoir sauvegarder des classe dans un fichier, il faut rappeler à cette méthode quelle est Serializable grâce à la ligne ajoutée "[Serializable()]" dans les classes BD\_Utilisateur, Utilisateur et Session.